This listing of claims replaces all prior versions, and listings of claims in the instant application:

Listing of Claims:

1.-29.   (Cancelled)

30.   (Currently Amended)   A small footprint device comprising:

at least one processing element, on said small footprint device, configured to execute groups of one or more program modules in separate contexts, wherein said separate contexts are included in a runtime environment on said small footprint device, and further wherein said runtime environment includes an operating system where said separate contexts are removed from and over said operating system on said small footprint device,

wherein said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions,

said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, and

each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context;

a memory, on the small footprint device, comprising instances of objects;

a context barrier, in said runtime environment and removed from and over said operating system, for

separating and isolating said contexts, said context barrier configured for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized; and

an entry point object, in said runtime environment and removed from and over said operating system, for permitting one program module, in one of said separate contexts, to directly access information from another program module, in another of said separate contexts, across said context barrier.

31.   (Previously Presented)   The small footprint device of claim 30 in which said context barrier allocates separate name spaces for each program module.

32.   (Previously Presented)   The small footprint device of claim 30 in which at least two program modules can access said entry point object even though they are located in different respective name spaces.

33.   (Previously Presented)   The small footprint device of claim 30 in which said context barrier allocates separate memory spaces for each program module.

34. (Previously Presented) The small footprint device of claim 33 in which at least two program modules can access said entry point object even though they are located in different respective memory spaces.

35. (Previously Presented) The small footprint device of claim 30 in which said context barrier enforces security checks on at least one of a principal, an object, and an action.

36. (Previously Presented) The small footprint device of claim 35 in which at least one security check is based on partial name agreement between a principal, and an object.

37. (Previously Presented) The small footprint device of claim 36 in which at least one program can access said entry point object without said at least one security check.

38. (Previously Presented) The small footprint device of claim 35 in which at least one security check is based on memory space agreement between a principal and an object.

39. (Previously Presented) The small footprint device of claim 38 in which at least one program can access said entry point object without said at least one security check.

40. (Previously Presented) The small footprint device of claim 30 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

41. (Previously Presented) The small footprint device of claim 30 wherein

said memory comprises object header data, said object
header data comprising information associated with at
least one of said instances of objects; and

said controlling execution is based at least in part
on said object header data.

42. (Previously Presented) The small footprint device of
claim 30 wherein

said memory is partitioned into a plurality of memory
spaces with instances of objects allocated for storage in
one of said plurality of storage spaces; and

said controlling execution is based at least in part
on determining the storage space allocated to an executing
object instance and an accessed object instance.

43. (Currently Amended) A method of operating a small
footprint device that includes a processing machine, wherein
program modules are executed on the processing machine, the
method comprising:

separating contexts, on said small footprint device,
using a context barrier, said context barrier configured
for controlling execution of at least one instruction of
one of zero or more sets of instructions comprised by a
program module based at least in part on whether said at
least one instruction is executed for an object instance
associated with a first one of said separate contexts and
whether said at least one instruction is requesting access
to an instance of an object definition associated with a
second one of said separate contexts, wherein said
separate contexts and said context barrier are included in
a runtime environment on said small footprint device and
further wherein said runtime environment includes an
operating system where said separate contexts and said

context barrier are removed from and over said operating system,

said separating further comprising:

preventing said access if said access is unauthorized; and

enabling said access if said access is authorized;

- executing groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context; and

permitting direct access to information from one program module, in one of said separate contexts, by another program module, in another of said separate contexts, across said context barrier using an entry point object wherein said entry point object is in said runtime environment and is removed from and over said operating system.

44.   (Previously Presented)   The method of claim 43 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

45.   (Previously Presented)   The method of claim 43 wherein said controlling execution is based at least in part on

object header data comprising information associated with at
least one of said instances of objects.

46.   (Previously Presented)   The method of claim 43
wherein

a memory of said small footprint device is
partitioned into a plurality of memory spaces with
instances of objects allocated for storage in one of said
plurality of storage spaces; and

said controlling execution is based at least in part
on determining the storage space allocated to an executing
object instance and an accessed object instance.

47.   (Currently Amended)   A method of permitting access to
information on a small footprint device from a first program
module to a second program module separated by a context
barrier, said small footprint device comprising:

at least one processing element, on the small
footprint device, configured to execute groups of one or
more program modules in separate contexts, said one or
more program modules comprising zero or more sets of
executable instructions and zero or more sets of data
definitions, said zero or more sets of executable
instructions and said zero or more data definitions
grouped as object definitions, each context comprising a
protected object instance space such that at least one of
said object definitions is instantiated in association
with a particular context wherein said separate contexts
are included in a runtime environment on the small
footprint device and further wherein said runtime
environment includes an operating system where said
separate contexts are removed from and over said operating
system;

a memory, on said small footprint device, comprising instances of objects; and

a context barrier, in said runtime environment and removed from and over said operating system, for separating and isolating said contexts, said context barrier configured for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized, the method comprising:

creating an entry point object, in said runtime environment and removed from and over said operating system, which may be accessed by at least two program modules; and

using said entry point object to permit direct access to information from one program module of said at least two program modules, in one of said separate contexts, by an other program module of said at least two program modules, in another of said separate contexts, across said context barrier.

48.   (Previously Presented)   The method of claim 47 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

49.   (Previously Presented)   The method of claim 47
wherein said controlling execution is based at least in part on
object header data comprising information associated with at
least one of said instances of objects.

50.   (Previously Presented)   The method of claim 47
wherein

a memory of said small footprint device is
partitioned into a plurality of memory spaces with
instances of objects allocated for storage in one of said
plurality of storage spaces; and

said controlling execution is based at least in part
on determining the storage space allocated to an executing
object instance and an accessed object instance.

51.   (Currently Amended)   A computer program product,
comprising:

a memory storage medium; and

a computer controlling element comprising
instructions for implementing a context barrier on a small
footprint device and for bypassing said context barrier
using an entry point object to permit direct access to
information from one program module, in one context, by
another program module, in a different separate context,
wherein said context barrier and said entry point object
are included in a runtime environment on the small
footprint device and further wherein said runtime
environment includes an operating system where said
context barrier and said entry point are removed from and
over said operating system, said small footprint device
comprising:

at least one processing element, on said small
footprint device, configured to execute groups of one or
more program modules in separate contexts, said one or
more program modules comprising zero or more sets of
executable instructions and zero or more sets of data
definitions, said zero or more sets of executable
instructions and said zero or more data definitions
grouped as object definitions, each context comprising a
protected object instance space such that at least one of
said object definitions is instantiated in association
with a particular context where said separate contexts are
included in said runtime environment and are removed from
and over said operating system;

a memory, on the small footprint device, comprising
instances of objects; and

a context barrier for separating and isolating said
contexts, said context barrier configured for controlling
execution of at least one instruction of one of said zero
or more sets of instructions comprised by a program module
based at least in part on whether said at least one
instruction is executed for an object instance associated
with a first one of said separate contexts and whether
said at least one instruction is requesting access to an
instance of an object definition associated with a second
one of said separate contexts, said context barrier
further configured to prevent said access if said access
is unauthorized and enable said access if said access is
authorized.

52.   (Cancelled)

53.   (Currently Amended)   A computer program product, comprising:

a memory storage medium; and

a computer controlling element comprising instructions for separating a plurality of programs on a small footprint device by running them in respective contexts and for permitting one program to access information from another program by bypassing a context barrier using an entry point object to permit direct access to information from one program, in one context, by another program in a different separate context, wherein said context barrier and said entry point object are included in a runtime environment on the small footprint device and further wherein said runtime environment includes an operating system where said context barrier and said entry point are removed from and over said operating system, said small footprint device comprising:

at least one processing element, on said small footprint device, configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context where said separate contexts are included in said runtime environment and are removed from and over said operating system;

a memory, on the small footprint device, comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

54. (Cancelled)

55. (Cancelled)

56. (Cancelled)

57. (Currently Amended) A method of transmitting code over a network, comprising transmitting a block of code from a server, said block of code comprising instructions for implementing an entry point object for bypassing a context barrier on a small footprint device over a communications link, wherein said context barrier and said entry point object are included in a runtime environment and further wherein said runtime environment includes an operating system where said context barrier and said entry point are removed from and over said operating system and further wherein said entry point object permits direct access to information from one program module, in one context, by another program module in another different context, said small footprint device comprising:

at least one processing element, on the small footprint device, configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context where said separate contexts are included in said runtime environment, on the small footprint device, and are removed from and over said operating system;

a memory, on the small footprint device, comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.